Boltzmann Equation for WIMP Dark Matter

Jacinto Paulo Neto

June 2024

The Boltzmann equation for WIMP Dark matter is given by

$$\frac{dY_\chi(x)}{dx} = -\frac{s(x)\langle\sigma v\rangle}{x\,H(x)}\left[Y_\chi^2(x) - Y_\chi^{\mathrm{eq}\,2}(x)\right] \tag{1}$$

where $x = m_\chi/T$, the equilibrium comoving number density

$$Y_\chi^{\mathrm{eq}}(x) = \frac{45}{4\pi^4}\frac{g_\chi}{g_{\star s}}x^2 K_2(x), \tag{2}$$

with g_χ corresponding to the dark matter degrees of freedom and $K_2(x)$ is the modified Bessel function of 2nd-kind. Moreover, the entropy density

$$s(x) = \frac{2\pi^2}{45}g_{\star s}m_\chi^3 x^{-3}, \tag{3}$$

and the Hubble rate

$$H(x) = \frac{\pi}{3}\sqrt{\frac{g_\star}{10}}\frac{m_\chi^2}{M_{\mathrm{P}}\,x^2}, \tag{4}$$

with $M_{\mathrm{P}} = 2 \times 10^{18}$ GeV. For the dark matter freeze-out $g_\star = g_{\star\,s} = 106.8$. On the next page, we exhibit a Python code to solve it numerically. The result must be equal or similar to Fig. 1.

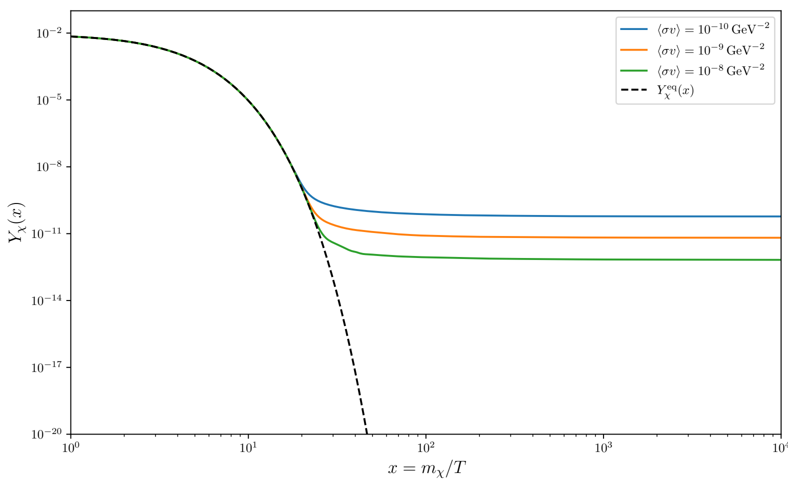

Figure 1: Numerical solution of the Boltzmann equation for WIMP Dark Matter.

```
import numpy as np
from scipy.integrate import solve_ivp
from scipy.special import kn
import matplotlib.pyplot as plt
from matplotlib import rc
rc('font', **{'family': 'serif', 'serif': ['Computer Modern']})
rc('text', usetex=True)
rc('text.latex', preamble=r'\usepackage{amsmath}')

# Functions
def s(x):
    return (2 * np.pi**2 / 45) * gstar * mchi**3 * x**-3

def H(x):
    return np.sqrt(np.pi**2 * gstar / 90) * mchi**2 / (Mp * x**2)

def Yeq(x):
    return (45 / (4 * np.pi**4)) * (gchi /gstar) * x**2 * kn(2, x)

# Boltzmann equation
def boltzmann_eq(x, Y, sigmav):
    Yeqx = Yeq(x)
    return -s(x) * sigmav / (x * H(x)) * (Y**2 - Yeqx**2)

# Constants
mchi = 100.0  # GeV
Mp = 2.435e18  # GeV
gchi = 4;  gstar = 106.8

```

```python
#  Annihilation cross-sections
sigmav_values = [1e-10, 1e-9, 1e-8] # GeV^{-2}

# Initial conditions and integration range
xinit = 1e-2; xend = 1e4; Y0 = Yeq(xinit)

# Solving the differential equation for each cross-section
xvalues = np.logspace(np.log10(xinit), np.log10(xend), 1000)
solutions = {}

for sigmav in sigmav_values:
    sol = solve_ivp(boltzmann_eq, [xinit, xend], [Y0], args=(sigmav,), dense_output=True, method='BDF', atol=1e-12, rtol=1e-12)
    Yvalues = sol.sol(xvalues)[0]
    solutions[sigmav] = Yvalues

# Plotting the solution
plt.figure(figsize=(10, 6))

for sigmav in sigmav_values:
    if(sigmav == 1e-10):
        plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle =10^{-10} \, {\rm GeV}^{-2}$')
    if(sigmav == 1e-9):
        plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle =10^{-9} \, {\rm GeV}^{-2}$')
    if(sigmav == 1e-8):
        plt.plot(xvalues, solutions[sigmav], label=r'$\langle \sigma v \rangle =10^{-8} \, {\rm GeV}^{-2}$')

plt.plot(xvalues, Yeq(xvalues), label=r'$Y_\chi^{\rm eq}(x)$', linestyle='--', color='black')
plt.xscale('log')
plt.yscale('log')
plt.xlabel(r'$x = m_\chi / T$', fontsize=14)
plt.ylabel(r'$Y_\chi(x)$',fontsize=14)
plt.xlim(1,1e4)
plt.ylim(1e-20,1e-1)
plt.legend()
plt.savefig('WIMPDM-BEQ.pdf')
plt.show()
```

Listing 1: Solution of the Boltzmann Equation for WIMP Dark Matter